Video serv r scheduling for simultaneous read-write requests

Patent Number: US6263411
Publication date: 2001-07-17
Inventor(s): AREF WALID G (EG); ALONSO RAFAEL (US); KAMEL IBRAHIM (US)
Applicant(s): MATSUSHITA ELECTRIC IND CO LTD (US)
Requested Patent: JP10162507
Application Number: US19960718279 19960920
Priority Number(s): US19960718279 19960920
IPC Classification: G06F12/00
EC Classification:
Equivalents: JP3372842B2

Abstract

A process is presented for supporting simultaneous disk read and write requests in a video server environment. Read requests are the result of movie viewing, while write requests are the result of video clip editing or movie authoring procedures. Due to real-time demands of movie viewing, read requests have to be fulfilled within certain deadlines, otherwise they are considered lost. Since the data to be written into disk is stored in main memory buffers, write requests can be postponed until critical read requests are processed. However, write requests still have to be proceeded within reasonable delays and without the possibility of indefinite postponement. This is due to the physical constraint of the limited size of the main memory write buffers. The new process schedules both read and write requests appropriately, to minimize the amount of disk reads that do not meet their presentation deadlines, and to avoid indefinite postponement and large buffer sizes in the case of disk writes

(19)日本国特許庁(JP)　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平１０－１６２５０７

(43)公開日　平成10年(1998)6月19日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ１１Ｂ	20/10	３０１	Ｇ１１Ｂ	20/10	３０１　Ｚ
Ｇ０６Ｆ	3/06	３０１	Ｇ０６Ｆ	3/06	３０１　Ｓ
Ｈ０４Ｎ	7/16		Ｈ０４Ｎ	7/16	Ａ

審査請求　未請求　請求項の数１８　ＯＬ　（全１５頁）

(21)出願番号　特願平9-255319

(22)出願日　平成9年(1997)9月19日

(31)優先権主張番号　08/718279

(32)優先日　1996年9月20日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者　イブラヒム・カメル
アメリカ合衆国08852ニュージャージー州モンマス・ジャンクション、サイプレス・コート3032番

(72)発明者　ワリド・ジー・アレフ
アメリカ合衆国95051カリフォルニア州サンタ・クララ、セント・イグナティウス・プレイス3282番　アパートメント311

(74)代理人　弁理士　青山　葆　（外1名）

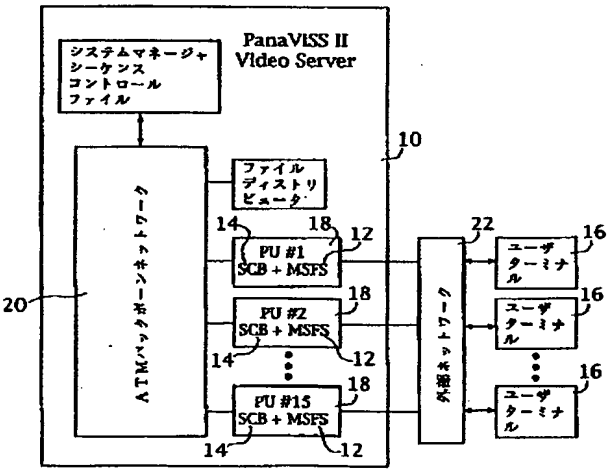
最終頁に続く

(54)【発明の名称】同時リードーライト要求用ビデオサーバスケジューリング

(57)【要約】

【課題】　ビデオ・オン・デマンド要求により同時にビデオオーサリング及び編集を処理できるように、ビデオサービスの機能拡張を意図したものである。

【解決手段】　ビデオサーバに於ける同時ディスクリード及びライト要求をサポートすると共に、バッファプールに於ける有用なスペース量に基づいてライト要求に対するデッドラインを割り当てるステップと；リード要求と共通ディスクキューに於ける前記要求を挿入するステップと；そして、キューの先頭に来る前記リード及びライト要求を処理するステップを有する方法を提供する。



1

【特許請求の範囲】

【請求項１】　ビデオサーバーでディスクリード要求とディスクライト要求を同時に保持する方法であって、
バッファープールの利用可能な空間の総量に基づく上記ライト要求にデッドラインを割り当て、
共通のディスクキューに上記リード要求とライト要求とを挿入し、
上記キューの先頭に来る上記リード要求と上記ライト要求を処理するもの。

【請求項２】　共通のディスクキューに上記リード要求及び上記ライト要求を挿入する上記ステップが、スキャンオーダーに上記ライト要求が従う位置を決定し、スキャンオーダーに上記ライト要求を挿入することで別のデッドラインが侵害されるかどうかを決定し、侵害されない場合、上記ライト要求をスキャンオーダーに上記ディスクキューに挿入するステップをさらに含んでいる請求項１記載の方法。

【請求項３】　共通のディスクキューに上記リード要求及び上記ライト要求を挿入する上記ステップが、新ライト要求がスキャンオーダーに挿入されることで、上記ライト要求のデッドラインが侵害されるかどうかを決定し、侵害される場合、上記キューの先頭に上記新しいライト要求を位置づけるステップをさらに含んでいる請求項２記載の方法。

【請求項４】　別のライト要求のデッドラインが侵害されたかどうかを決定し、別のライト要求のデッドラインが侵害されている場合、侵害されたデッドラインを有する上記ライト要求をスキャンオーダーの上記キューの先頭に移動させるステップをさらに含んでいる請求項３記載の方法。

【請求項５】　スキャンオーダーに上記新ライト要求を挿入することで、別のリード要求が侵害されるかどうかを決定し、侵害された場合、上記キューの終端部に上記新ライト要求を挿入させることで、上記新ライト要求のデッドラインが侵害されるかどうかを決定し、侵害されない場合、上記キューの終端部に上記新ライト要求を挿入するステップをさらに含んでいる請求項３記載の方法。

【請求項６】　上記新ライト要求のデッドラインを侵害せずに、上記新ライトを上記キューの終端部に挿入することができないと決定した場合、スキャンオーダーに上記新ライト要求を挿入することで、別のライト要求のデッドラインが侵害されるかどうかを決定し、侵害されない場合、上記新ライト要求がスキャンオーダーに挿入される請求項５記載の方法。

【請求項７】　スキャンオーダーに上記新ライト要求を挿入することで、別のライト要求のデッドラインが侵害するとを決定された場合、侵害されたデッドラインを有する全ライト要求が、スキャンオーダーの上記キューの先頭に移動する請求項６記載の方法。

2

【請求項８】　共通のディスクのキューに上記リード要求と上記ライト要求とを挿入するステップは、スキャンオーダーに新リード要求を挿入することで、上記キューのリード要求又はライト要求のデッドラインが侵害されるかどうかを決定し、侵害されない場合、上記スキャンオーダーに上記新リード要求を挿入するステップをさらに含んでいる請求項１記載の方法。

【請求項９】　共通のディスクのキューに上記リード要求と上記ライト要求とを挿入するステップは、スキャンオーダーに新リード要求を挿入することで、上記キューのリード要求又はライト要求のデッドラインが侵害されるかどうかを決定し、侵害されている場合、上記新リード要求のデッドラインが侵害されているかどうか、上記新リード要求が上記キューの終端部に上記新リード要求が挿入されているかどうか決定し、挿入されていない場合、上記キューの終端部に上記リード要求を挿入するステップをさらに含んでいる請求項１記載の方法。

【請求項１０】　上記ライト要求にデッドラインを割り当てる上記ステップは、バッファープールのフリーブッファースロットの数量、及び上記システムのディスクライト要求の到着比率の関数として上記デッドラインを計算するステップを含んでいる請求項１記載の方法。

【請求項１１】　ディスクリード要求とディスクライト要求を同時に保持するビデオサーバー構成であって、
メモリーバッファープールであって、上記リード要求及び上記ライト要求が一時的に貯蔵されるもの、
共通のディスクキューであって、上記メモリーバッファーからの上記リード要求及び上記ライト要求が受け入られるもの、
上記リード要求とライト要求を処理するファイルサーバーであって、すくなくともディスクを１つ保持するもの、
上記ディスクキューに上記リード要求及びライト要求を挿入する位置を決定するコントロール手段であって、上記ライト要求が割り当てられたデッドラインであるものを含んでいるもの。

【請求項１２】　新ライト要求がスキャンオーダーにゆだねられる位置を決定することを含んでいる処理に従って、上記リード要求及び上記ライト要求が上記キューに挿入される請求項１１記載のビデオサーバー。

【請求項１３】　さらに上記処理が、スキャンオーダーに上記新しいライト要求を挿入することでデッドラインが侵害されたかどうかを決定し、侵害されていない場合、スキャンオーダーに上記新ライト要求を挿入する請求項１２記載のビデオサーバー。

【請求項１４】　さらに上記処理は、上記新ライト要求がスキャンオーダーに挿入されている場合、上記新ライト要求のデッドラインが侵害されているかどうかを決定し、侵害されている場合、上記新ライト要求を上記ディスクキューの先頭に挿入する請求項１３記載のビデオサ

ーバー。

【請求項１５】　さらに上記処理は、上記ディスクキューの先頭に上記新ライト要求が挿入されることで、別のライト要求のデッドラインが侵害されたかどうかを決定して、侵されている場合、侵害されているデッドラインを有する上記ライト要求が、スキャンオーダーの上記キューの先頭に移動される請求項１４記載のビデオサーバー。

【請求項１６】　さらに上記処理は、上記新ライト要求がスキャンオーダーで挿入されている場合に、上記新ライト要求のデッドラインが侵害されているかどうかを決定し、侵害されていない場合で、上記新ライト要求がスキャンオーダーに挿入された場合、リード要求のデッドラインが侵害されたかどうかを決定し、侵害されている場合は、上記新ライト要求のデッドラインを侵害することなしに、上記新ライト要求がディスクキューの終端部に挿入することが可能かどうかを決定し、可能である場合、上記ライト要求が上記ディスクキューの終端部に挿入される請求項１５記載のビデオサーバー。

【請求項１７】　上記処理は、上記新ライト要求がスキャンオーダーに挿入されている場合に、上記ライト要求のデッドラインが侵害されているかどうかを決定し、デッドラインが侵害されている場合、侵害されているデッドラインを有する全ライト要求を上記ディスクキューの先頭に移動させる請求項１２記載のビデオサーバー。

【請求項１８】　上記コントロール手段が、各々の上記ライト要求にデッドラインを、上記メモリーバッファープールのフリーバッファースロットの数量とシステムのディスクライト要求の到達比率との関数として割り当てる請求項１２記載のビデオサーバー。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、同時リード及びライト要求に関連するリアルタイム要求に於いて、ユーザによって成される同時リードーライト要求をサポートする新しいディスクスケジューリングアルゴリズムに関する。

【０００２】

【従来の技術】ビデオ・オン・デマンド及びビデオオーサリングツールが、非常な興味と共にマルチメディアアプリケーションを切り開くものとして出現している。これらは、高帯域幅を必要とすると共に、リアルタイム要求を擁する特別なハードウェアとネットワーキングプロトコルを必要とする。ビデオ・オン・デマンドアプリケーションを取り扱う各種のビデオサーバアーキテクチャが提案されている。１９９４年５月発行の第５回インターナショナルワークショップオンマルチメディアコミュニケーション誌、３４１－３４６頁に発表されたＹ．イトー及びＴ．タナカによる「ＡＴＭスイッチング技術を用いたビデオサーバ」；１９９２年発行のＡＣＭ会報のコンピュータシステム誌の１０（４）、３１１－３３７頁に発表されたディビッドアンダーソン、ヨシヒトオサワ、及びラメシュゴヴィンダンによる「連続メディアのファイルシステム」；１９９２年７月発行のＩＥＥコミュニケーションマガジン、５６－６４頁に発表されたＰ．Ｖ．ランガン、Ｈ．Ｍ．ビン、及びＳ．ラマナサンによる「オン・デマンドマルチメディアサービスの設計」がある。本発明は、第５回インターナショナルワークショップオンマルチメディアコミュニケーション誌に提案された、ＡＴＭスイッチにより内部接続されるファイルサーバ及び複数のディスクを用いるようなサーバアーキテクチャに良く適合する。本発明は、その他のアーキテクチャにも実施できる。数種のパフォーマンススタディ及びシミュレーションによって、本システムの性能が見積もられている。

【０００３】ビデオサーバは各種のアプリケーションをサポート出来る。これらのアプリケーションの中には、ビデオ・オン・デマンド、及びビデオオーサリング或いは編集が有る。各アプリケーションは、それ自身のシステム要求を有している。本発明は、そのようなシステムのディスクスケジューリング要求に主眼を置いている。例えば、ディスクと言う観点から見ると、ビデオ編集アプリケーションがリード及びライトの両要求を発行している間に、ビデオ・オン・デマンドアプリケーションはディスクに対してリードオンリー要求を発行する。更に、ある種のアプリケーションは、リアルタイムデッドラインを所有或いは非所用するリード及／或いはライト要求がデッドラインの前に使用されるようにこれらの要求を発行する。例えば、ビデオ・オン・デマンドに置いては、各リード要求は、所定のデッドライン内に実行されねばならない。最終的に、ある種のアプリケーションは、自身のリード或いはライト要求の幾つかがシステムによって喪失されるのを、つまりディスクによって利用したり実行されたりしないことを容認する。例えば、フレームの幾つかは、ディスクに於ける渋滞の為に、ビデオ表示中に喪失され得る。

【０００４】正式には、ディスクスケジューリングの観点から、リード及びライト要求を、それぞれ特定のアプリケーションに対して有用で且つ互いに異なる要求を有する４つのカテゴリーに分類するものとする。これらのカテゴリーには：

１．ｄ１要求：　これらは、渋滞の場合には喪失（lost）し得る要求とデッドライン（deadline）を有するリード或いはライト要求である。本カテゴリーのリード及びライト要求は、それぞれ、R_{d1}及びW_{d1}として参照される。

２．ｄｎ要求：　これらは、システムの渋滞に関わらず喪失が許されない（not）要求とデッドライン（deadline）を有するリード或いはライト要求である。本カテゴリーのリード及びライト要求は、それぞれ、R_{dn}及びW

$_{dn}$として参照される。

３．ｎ１要求：　これらは、渋滞の場合には喪失（lost）が許される要求を有するがデッドラインを有しない（no）リード或いはライト要求である。本カテゴリーのリード及びライト要求は、それぞれ、R_{n1}及びW_{n1}として参照される。

４．ｎｎ要求：　これらは、システムの渋滞に関わらず喪失が許されない（not）要求を有するがデッドラインを有しない（no）リード或いはライト要求である。本カテゴリーのリード及びライト要求は、それぞれ、R_{nn}及びW_{nn}として参照される。

【０００５】例えば、ビデオ・オン・デマンドに於けるアプリケーションが、R_{d1}要求だけをサポートしなければならないディスクスケジューリングプロセスで有る場合である。つまりデッドラインを有するが、そのデッドラインの前に要求が使用されない場合には喪失が可能なリード要求である。要求の各カテゴリーに対して、異なるディスクスケジューリングプロセスをデザインする必要がある。カテゴリーｎ１（R_{n1}或いはW_{n1}）に所属する要求に対しては、デッドラインが無いので、これらの要求が満足されるまで遅延できるので常に真である。それ故に、それらが用いられるまで待たせることができるように、それらを喪失させることはシステムにとって意味が無い。結果的に、W_{n1}及びR_{n1}は此処では、それぞれ、W_{nn}及びR_{nn}として扱われる。なぜならば、それらはデッドラインを有しないし、決して喪失されないからである。結果として、此処ではｄｌ、ｄｎ、及びｎｎカテゴリーに対してのみ、ディスクスケジューリング技法を考慮する。

【０００６】

【発明が解決しようとする課題】リード要求に関するR_{d1}及びR_{nn}カテゴリーと、ライト要求に関するW_{nn}カテゴリーとをサポートする新しいディスクスケジューリングプロセスを提供する。更に、これらのカテゴリーの組み合わせて、ビデオサービス用のある種の共通アプリケーション自然に整合する方法及び装置を提供することを目的とする。

【０００７】

【課題を解決するための手段】本発明は、前記課題を解決するための手段として、である。ビデオサーバーでディスクリード要求とディスクライト要求を同時に保持する方法であって、バッファープールの利用可能な空間の総量に基づく上記ライト要求にデッドラインを割り当て、共通のディスクキューに上記リード要求とライト要求とを挿入し、上記キューの先頭に来る上記リード要求と上記ライト要求を処理する方法を提供する。更に、ディスクリード要求とディスクライト要求を同時に保持するビデオサーバー構成であって、メモリーバッファープールであつて、上記リード要求及び上記ライト要求が一時的に貯蔵されるもの、共通のディスクキューであって、上記メモリーバッファーからの上記リード要求及び上記ライト要求が受け入られるもの、上記リード要求とライト要求を処理するファイルサーバーであって、すくなくともディスクを１つ保持するもの、上記ディスクキューに上記リード要求及びライト要求を挿入する位置を決定するコントロール手段であって、上記ライト要求が割り当てられたデッドラインであるものを含んだ構成を提供する。

【０００８】

【発明の実施の形態】本発明は、ビデオプリゼンテーション及びビデオ編集に対するリアルタイム要求に於いて、同時リード及びライト要求を処理するディスクスケジューリングングに関する。更に詳述すれば、本発明のプロセスは、ライト要求に対するW_{nn}カテゴリーとリード要求に対するR_{d1}及びR_{nn}カテゴリーをサポートする。ビデオ編集がリード及びライトの両要求（R_{nn}及びW_{nn}要求）に至るのに対して、ビデオプリゼンテーション或いは表示はファイルシステムに対するリード要求（R_{d1}）に至る。

【０００９】本発明は、本明細書の一部として取り込まれる１９９４年５月発行の第５回インターナショナルワークショップオンマルチメディアコミュニケーション誌、３４１－３４６頁に発表されたＹ．イトー及びＴ．タナカによる「ＡＴＭスイッチング技術を用いたビデオサーバ」に提案されたビデオサーバアーキテクチャに置いて実施することが出来る。本アーキテクチャに於ける主な要素が図１に示されている。サーバアーキテクチャの概要を以下に述べる。

【００１０】ビデオサーバ１０はＭＰＥＧ-エンコード（圧縮）されたビデオストリームをサポートしている。各ストリームは、メディアセグメントファイル（ＭＳＦ）ブロックと呼ばれる固定長単位に分解される。所定のビデオストリームに対するメディアセグメントファイルブロックは、ファイルシステムの全体に分散して保存される。ファイルシステムは複数のディスク保存サーバを有しており、そこでは各保存サーバがメディアセグメントファイルサーバ（ＭＳＦＳ）と呼ばれる。メディアセグメントファイルサーバ１２は、様々なビデオストリームに所属するメディアセグメントファイル（ＭＳＦ）のブロックを保存する。ビデオストリームを正しい順番で取り出す為に、シーケンスコントロールブローカ（ＳＣＢ）１４は、ビデオストリームの全メディアセグメントファイルブロックに対するポインタの順番リストを保存する。シーケンスコントロールブローカ１４は、ユーザの代わりに、ビデオ再生ストリームを維持するべく作用する。各シーケンスコントロールブローカ１４は、例えば２４或いは６４ユーザというような一以上のユーザターミナル１６を同時にサポート出来る。一つのシーケンスコントロールブローカ１４に接続されるユーザターミナル１６の数は、システム全体として全てのユーザに

対して連続ビデオ再生を保証できるように予め決められる。

【００１１】ビデオ再生セッションの初期化では、シーケンスコントロールブローカ（ＳＣＢ）１４が、要求された映画のポインタリストを取り出す。再生中には、シーケンスコントロールブローカ１４が、ユーザの代わりにメディアセグメントファイルサーバ（ＭＳＦ）１２にリード要求を送出して、ユーザに不断のサービスを保証する。シーケンスコントロールブローカ（ＳＣＢ）１４は、更に、例えば早送りとか巻き戻しという仮想ビデオカセットレコーダ（ＶＣＲ）機能要求処理も受け持つ。

【００１２】シーケンスコントロールブローカ（ＳＣＢ）１４及びメディアセグメントファイル（ＭＳＦ）セクタＳ１２は、一つのブロックに組み込まれている。ビデオサーバに於いては、例えば１５個という、複数のＰＵユニット１８が存在する。各シーケンスコントロールブローカ（ＳＣＢ）１４は、対応するメディアセグメントファイルサーバ１２に存在するデータに直接アクセスできる。しかしながら、メディアセグメントファイルは複数のメディアセグメントファイルサーバ１２にまたがって保存されているので、シーケンスコントロールブローカ１４はその他のプロセッシングユニット１８のメディアセグメントファイルサーバ１２にアクセスする必要がある。

【００１３】プロセッシングユニット１８が通信する為に、全てがＡＴＭスイッチ２０に接続される。例えば、シーケンスコントロールブローカ１４がＡＴＭスイッチ２０を使って、複数のプロセッシングユニットに存在するメディアセグメントファイルサーバ１２に渡るメディアセグメントファイル（ＭＳＦ）を取り出す。ＡＴＭスイッチ２０は、メディアセグメントファイルサーバ１２の全てに対して、システムに於ける各シーケンスコントロールブローカ１４からの接続を保証するメッシュネットワークを提供する。

【００１４】シーケンスコントロールブローカ１４は、他の側から、ビデオサーバ１０をエンドユーザ１６に接続する外部ネットワーク２２に接続されている。ユーザ１６はセット・トップボックスを用いて外部ネットワーク２２に接続される。このセット・トップボックスは、ＭＰＥＧエンコードされたビデオデータをデコードし、仮想ＶＣＲ機能要求に対するユーザインターフェースを提供し、シーケンスコントロールブローカ１４と通信する機能を有している。

【００１５】R_{d1}リード要求は、特定の時間に特定の映画或いはビデオクリップを見るというユーザのデマンドの結果である。一旦システムに入れば、ユーザには特定の品質のサービスが保証される。このことは、聴視者が気づかない程フレームのロス率が非常に小さいフレームの連続ストリームを聴視するということで表される。フレームロスは多岐に渡る理由により起こり得るが、此処ではメディアセグメントファイルサーバ１２中のディスクの渋滞に起因する喪失を対象とする。メディアセグメントファイルサーバ１２のディスクに対する各リード要求は、リアルタイムデッドラインを有している。メディアセグメントファイルサーバ１２は、デッドライン中に要求を実行できるか否かを判定する。これに基づいて、メディアセグメントファイルサーバ１２は要求を受諾或いは拒絶する。後者の場合、拒絶されたリード要求に対応する頁は喪失されたと見なされる。

【００１６】シークタイムは、ディスクから頁を引き出す際に最も時間を要する部分である。ディスクスケジューリングプロセスの目的の一つは、ディスクから頁を引き出すことである。これは、シークタイムが最小になるようにディスクアクセス要求をキューすると共にオーダーすることにより達成出来る。プロセスＳＣＡＮは、この問題を処理する従来のディスクスケジューリングプロセスの一つである。Ａ．シルバーシャッツ及びＰ．Ｂ．ガルヴィン。アディソンウェズリィにより、１９９４年発行のオペレーティングシステムコンセプト、第４版に発表されている。ＳＣＡＮに於いては、ディスクヘッドは一方向（内側或いは外側）へ移動し、シリンダーポジションがディスクヘッドが現在位置している場所になるようなディスク要求を出す。ディスクヘッドは、シリンダーポジションの最高位置に達すると、移動方向を反転して、そのパスに沿って広がるディスク要求を出す。

【００１７】リアルタイム要求により、ＳＣＡＮは、要求された頁のリアルタイムコンシダレーションに適合するように修正されている。プロセスＳＣＡＮＲＴ（リアルタイムコンシデレーションを伴うＳＣＡＮ）は、ＳＣＡＮの修正版である。ディスク問い合わせ中に新たに挿入された要求がキュー中に既に存在するディスク要求のリアルタイム要求を破らない限り、ＳＣＡＮＲＴはディスク要求をＳＣＡＮの順番に出す。ＳＣＡＮの順番に挿入された新たな要求が、ディスクキューに既に存在する他の要求をデッドラインを損なわせる場合には、新たな要求はキューの終端で挿入される。この時には、新たに挿入されたディスク要求が、自身のデッドラインに間に合うように適時出すことが出来るかが判断される。もし間に合わないと判断される場合には、要求は無視され、喪失したものと見なされる。

【００１８】ｎｎ要求に対するｄ１要求の量、例えば混合比が５０％対５０％或いは９０％対１０％、によってシステム負荷特性が決定されるということ重要なことである。これは、ビデオサーバ１０中に於けるビデオプリゼンテーション対編集活動量を反映するものである。ビデオ編集或いはオーサリングセッションに於いて、リード及びライト要求を出すことができる。先ず、ライト要求について述べた後に、リード要求について述べる。

【００１９】ビデオ編集セッションに於いて、セッション中に出されるライト要求が、W_{nn}カテゴリーに属する

もモデルとして説明する。ライト要求に対するW_{nn}カテゴリーは、ビデオプリゼンテーションに対するリード要求と異なる特性を有しており、それ故にシステムによって異なるように扱われる。これらの特性は以下の如く要約出来る。

1．ディスクに書き込まれる任意の頁は、既にメインメモリライトバッファプール28中に予め保存されている。

2．W_{nn}ライト要求はリアルタイムデッドラインを有しない。それ故に、リード要求に比べて長時間の遅延を可能とする。

3．W_{nn}ライト要求はデッドラインを有していないが、不定時間の待機は出来ない。ライトバッファプール28が一杯になるのを防ぐ為に、将来のいつか実行されるべきである。

4．W_{nn}ライト要求は、システム負荷等によって喪失出来ない。ソステム負荷に関わらず、ライト要求は或る時間にシステムによって実行されねばならない。

5．システム負荷に基づいて、ライトバッファプール28は一杯になることが出来る。バッファスペースを新たに到着するライト要求の為に空けてやるために、この時点でペンディング中のW_{nn}ライト要求のいくらかは、図2に示すようにディスク30にフラッシュされねばならない。

6．W_{nn}ライト要求は、例えば電源不全のようなシステムクラッシュに対して柔軟で無ければ成らない。言い換えれば、ユーザの観点からは、システムが一旦ライト要求を認識したら、ユーザは書き込まれた頁P_wは将来の任意の時間にシステム中に存在していることを期待する。これは、P_wがメモリバッファ中に一次的に保存され得ると言う事実に反しているので、電源不全の時には、それが失われる可能性がある。システムはこのようなシナリオの事態を防止するべく必要な予防策を講じなければならない。

【0020】上述の特性点に基づいて、システムの性能にとって重要なパラメータの幾つかについて述べる。これらの特徴点の中で最も重要なのは、図2に示すように、メインメモリライトバッファプール28の大きさである。ライト要求によりシステムに化された要求は、より大きなライトバッファプールによって低減できると予期される。R_{dl}リード及びW_{nn}ライトの両要求を同時に取り扱う為に、システムは、同両要求が適合すると共に両タイプの要求をサポートする均質フレームを採用する。

【0021】本発明は、これを、上述したライト要求の特性に反映したライト要求W_{nn}の達成のために、人為的なデッドライン(区切り)を発展させることにより達成する。リード要求R_{dl}もまたデッドラインを持つので、本発明は、リードおよびライト要求の双方のカテゴリーに対する同質な処理方法を提供する。本発明のディスクのスケジュール作成技術を詳しく紹介する前に、いくつかの基本概念の概要を開示する。図5（A）および5（B）（集合して図5と呼ぶ)を参照すると、図5（A）は、通常のディスクのスケジュール作成のプロセスがいかに実施されるかを示す。図5（B）は、いかに本発明が動作するかを示す。これらの二つの図を比較すると、本発明のいくつかの原理を理解できる。

【0022】図5（A）において、分離したリードキー100とライトキュー102は、リードおよびライトのアクセスに役立つ。図5（A）において、ディスクは、104にて図形的に描かれている。個々のリード要求106は、それぞれ関連するデッドライン108を持つ。図5（A）において、個々のリード要求は、図式的に楕円106で示され、それらは関連するデッドラインのパラメータ108を持つ。これらのリード要求は、種々の異なった時間で、システムの一人または多くのユーザーから生じたものである。図5（A）でわかるように、リード要求は、デッドラインのパラメータに基づきキー100内のメモリ位置に割り当てられる。この結果、リード要求は、短いデッドラインでもってキューの先頭に接近して位置し、これにより、それらの要求はより迅速に処理される。そのデッドラインの属性は、リード要求を“エレベータ”の順に配列するために用いられる。その要求は、ディスクへの効率よいアクセスとなるような観点に従って配置される。その結果、リード要求は、可能な限り、蓄積された記録を連続的にリードできるように配列される。

【0023】そのリード要求に反して、ライト要求（図式的に110で示す）は割り当てられたデッドラインを持たない。この結果、システムのユーザーがライト要求を出したとき、その要求はライトキュー102内の次に利用可能なメモリ位置に置かれる。ディスク104へのリードおよびライト信号は、最初に処理用のリードキュー100によって処理され、リードキューをサポートするために使用されないフリー時間があるときのみ、処理用のライトキュー102によって処理される。

【0024】図5（B）を参照すると、本発明は、個別のリードおよびライトキューを利用するための必要性を省く。その替わりに、本発明は、単一の同質のキュー112を用い、これは、同じメモリ構成でレード要求とライト要求の双方をサポートする。上述したように、リード要求106は、関連したデッドラインの属性108を持つ。本発明は通常の実行とは異なり、すべてのライト要求に人為的なものではなく実際のデッドラインを割り当てる、優先モジュール114を採用する。その結果、図5（B）において、ライト要求110は、それぞれ関連したデッドラインの属性116を持つようになる。以下により詳細に述べるように、それらのデッドラインは、利用可能なフリーのバッファスペースに基づき割り当てられる。すべてのライト要求に人為的に割り当てら

れたデッドラインを持つことにより、リードおよびライト要求は、エレベータ式の位置決めスキームを用いて単一の同質のキュー１１２を通じて処理されるようになる。その構成は、バッファのオーバーフローを防止するために必要とされたとき、リード要求を犠牲にすることにより、ライト要求は失われない。実際に、本発明は、リードおよびライト要求がシステムに出されたとき、より優先順位の高いリードおよびライト要求をキューの先頭により接近させることにより、すべてのデッドラインを試した。

【００２５】次のパラメータは、本発明のプロセスを記述するために用いられる。

N_b：ライトバッファプール２８のバイトサイズ

P_w：ライトページのバイトサイズ（これらの二つのパラメータからN_w（バッファが収容できるライトページの数）を演算できる）

λ_w：ディスクライト要求のシステムへの到着速度

時刻ｔにおいて、ユーザーがページ、つまりP_wがディスク３０にライトされることを要求したと仮定する。ページP_wがディスク３０にデッドラインの前にライトされるように、ページP_wに対してデッドラインが割り当てられる。

【００２６】ライト要求のデッドラインは、次のようにして演算される。$n_w(t)$を時刻ｔにバッファ２８に存在するライト要求の個数、$n_f(t)$を時刻ｔにバッファプール２８内のフリーバッファスロットの個数とする。すると、$n_f(t)=N_w-n_w(t)$。最悪ケースの想定では、リード要求R_{d1}のため、バッファ２８からディスク３０へページは書き込まれず、同時に、新しいライト要求がλ_wの速度でライトバッファプール２８に続いて到着する。その結果、（新しいライト要求により）その時刻で、ライトページp_wがバッファプール２８に到着し、この最悪ケースの想定に対して、必要とされる時刻、つまり、ライトバッファプール２８が満杯になる前のｄ(ｔ)が予測される。ｄ(ｔ)は以下のように演算される。

$d(t)=t+n_f(t)/\lambda_w$

【００２７】そのｄ(ｔ)は、実際に、時刻ｔ、つまりｄ(ｔ)でのライトバッファプール２８内のすべてのページのデッドラインは、ライトバッファ２８内の現行のすべてのページに対する全体的なデッドラインであることに注目する。その結果、ページp_wがディスク３０に物理的に書き込まれたとき、バッファプール２８内で一つのバッファスロットをフリーにする。その結果、バッファ２８が満杯になるまでにより多くの時間があるので、バッファプール２８内のすべてのページに対するデッドラインｄ(ｔ)は緩和される。それ故、ｄ(ｔ)は次のようにして緩和される。t_oをｄ(ｔ)が修正されたラストの時刻とすると、

$d(t)\leftarrow t-t_o+d(t_o)+1/\lambda_w$

【００２８】これは、上記のｄ(ｔ)に対する公式と一致する、つまり、

$d(t)\leftarrow t-t_o+d(t_o)+1/\lambda_w$

$=t-t_o+t_o+n_f(t_o)/\lambda_w+1/\lambda_w$

$=t+(n_f(t_o)+1)/\lambda_w$

$=t+n_f(t)/\lambda_w$

t_oはバッファプール２８内でのあらゆる変化のラストタイムなので、$n_f(t)=n_f(t_o)+1$は、ライトページがディスク３０に物理的に書き込まれた後の新しいスペース合計である。

【００２９】ライトページが到着を要求したとき、対応するページ、つまりp_wは、ライトバッファプール２８内に位置する。ライト要求をディスクキュー３２内へ挿入するために、リード／ライトのスケジュール作成システムは、p_wに対するデットライン、つまり$d_w(t)$を割り当て、その$d_w(t)$は、ライトバッファプール２８の全体のデットラインｄ(ｔ)デットラインに対応する。つまり、$d_w(t)\leftarrow d(t)$。最後に、p_wおよびそのデットライン$d_w(t)$は、ディスクキュー３２に挿入される。本発明の主な利点は、スケジュール作成のプロセスが、ライト要求をスキャン順に、リード要求を扱うＳＣＡＮＲＴと同様な方法によって位置させるように試みる。このことが達成され、又、ディスクキュー３２内でリードとライトの要求の間で相互作用する機構を以下に述べる。

【００３０】先に述べたように、ＳＣＡＮＲＴディスクのスケジュール作成プロセスは、キュー、ディスク・スケジューラー・キュー、または単にディスクキューを保ち、ディスクキュー３２内への加入は、ディスク３０に１ページのリードまたはライトするための要求を示す。ディスクキュー３２での要求は、シークタイムを減じて、ディスクキュー３２内であらゆる要求に対してデッドラインの違反がないように、ＳＣＡＮＲＴ順に出される。リードのみに対して、ディスクキュー３２内のR_{d1}およびW_{nn}の同時要求を扱うとき、考慮すべき必要な４つの項目がある。これらは次のように分類される。

（１）ライト要求W_{nn}をディスクキュー３２に挿入し、

（２）リード要求R_{d1}をディスクキュー３２に挿入し、

（３）リード要求R_{d1}を処理し（ディスク３０から実際に読み出す）、（４）ライト要求W_{nn}を処理する（ディスク３０に実際に書き込む）。新しいプロセスは以下に述べるように、それらの各項目で異なる動作を実行する。

【００３１】ライト要求W_{nn}を挿入

図３に示したフローチャートに関しては、デイスクキュー３２へのライト要求W_{nn}の挿入が記載されている。ページp_wに対するライト要求が一旦、ライトバッファプールに到着すると（Ｓ４０）、メディアセグメントファイルサーバー１２は、上述したように、ライト要求にデッドラインを割り当てる（Ｓ４２）。次にライト要求がデイスクキュー３２に挿入される。これは以下のようにして達成される。

【００３２】メディアセグメントファイルサーバー１２のスケジュール作成プロセスは、スキャン順４４下においてページp_wがどこに行くかを決定し、そして、新しいライト要求をデイスクキュー３２内においてそのスキャン順の中に挿入する。このケースにおいていくつかの可能性が提示される。

１．デイスクキュー３２内でいずれのリードおよびライト要求のデッドラインが無く、挿入された新しいライト要求を含むとき、違反する。

２．新しいライト要求がスキャン順に挿入されるならば、オーダーのデッドラインは違反しない。しかし、挿入時、いくつかの他のリード要求のデッドラインが違反する。

３．新しいライト要求がスキャン順に挿入されるならば、オーダーのデッドラインは違反しない。しかし、挿入時、いくつかの他のライト要求のデッドラインが違反する。

４．新しいライト要求がスキャン順に挿入されるならば、そのライト要求のデッドラインが違反する。

【００３３】スケジュール作成プロセスがこれらの各々の可能性をいかに処理するかは以下に述べる。ケース１は、処理が容易で最も可能性の高いシナリオである。もしスケジュール作成プロセスが、新しいライト要求のデッドラインのみならず、キュー３２に既に存在するリードおよびライト要求のデッドラインに違反することなく、をキュー３２内へスキャン順にライト要求を挿入できることがみいだされたならば(Ｓ４６)、新しいライト要求は、そのスキャン順にデイスクキュー３２へ挿入される。

【００３４】ケース２において、新しいライト要求がスキャン順に挿入されたとき、新しいライト要求のデッドラインが違反しないが(Ｓ５０)、その挿入時、いくつかの他のリード要求のデッドラインが違反する(Ｓ５２)。このプロセスは、違反したデッドラインを持つリード要求の喪失を避けるように試みる。もし、ライト要求のデッドラインが違反することなく、そのライト要求がキュー３２の終端で置き換えることができるならば(Ｓ５４)、そのプロセスは、スキャン順ではなく、デイスクキュー３２の終端に新しいライト要求を位置させる。このことは、違反したデッドラインを持つリード要求が喪失するのに役立つ。ここでの最も高い可能性は、キュー３２の終端で別のスキャン順(次スキャン順と呼ぶ)を維持し、そして、新しいライト要求を単にキュー３２の終端に位置させる代わりに、その新しいライト要求を次スキャン順に挿入することである。

【００３５】ケース１およびケース２は、通常のシステム動作において典型的なシナリオを構築する。他方、ケース３および４は、ライト要求の速度がシステムの基本タイムで定められた上限を超過する極端なケースではある。システム動作の間に、ケース３およびケース４の発生回数が重要になるとき、要求パラメータが変化し、評価システムのパラメータを、例えばバッファプール２８のサイズを拡張を示す。

【００３６】スケジュール作成プロセスがいからケース３および４を扱うかを示すことに留まっており、新しいライト要求は、デイスクキュー３２内に既に存在する別のライト要求のデッドラインを違反させるために決定される(Ｓ５８)。この場合、そのプロセスは、違反したデッドラインを持つすべてのライト要求をデイスクキュー３２の先頭に移動する(Ｓ６０)。この結果、デッドラインが違反するようになったリード要求は、喪失したと考えられる。もしライト要求のデッドラインが違反したとき、新しいライト要求はスキャン順に挿入される(Ｓ５９)。

【００３７】ケース３(新しいライト要求が他のライト要求のデッドラインを違反させる)は、ライトバッファプール２８が混雑し、システムがオーバーロードしたことを示す。この状況は、ビデオサーバ１０の許可制御プロセスにより避けなければならない。ケース３では、新しいライト要求がそのスキャン順に挿入されるならば、新しいライト要求のデッドラインが違反する(Ｓ５０)。ライト要求のデッドラインのリコールは、ライト要求が満杯になる時刻に基づき演算される。その結果、ライト要求は、その違反したデッドラインを持つ余裕がなく、この結果、バッファがオーバーフローし、ライト要求の喪失が起こり得る。W_{nn}ページを喪失することはできないので(定義により)、プロセスは、新しいライト要求をそのデッドラインを違反させることなくスケジュールしなしてはならない。これは以下のようにして達成される。

【００３８】そのプロセスは、新しいライト要求をキュー３２の先頭に設ける(Ｓ６２)。デッドラインが違反するようになったリード要求は、喪失したとみなされるが、デッドラインが違反するようになったライト要求は(Ｓ６４)、ケース３として扱われる、つまり、新しいライト要求に沿ってキュー３２の先頭に移動される。ケース３およびケース４の双方で、キュー３２の先頭に移動したライト要求は、スキャン順に自身の間で順位が決められる。絶対デッドラインがリードおよびライト要求の双方に格納されるため、新しい要求の挿入時、スキャン順において新しい要求以前の各要求のデッドラインはアップデートされる必要はないことに注目。

【００３９】ライト要求W_{nn}の処理

上述のすべてのケースにおいて、ライトページp_wが一旦、ディスクキュー３２に挿入されると、そのライトページはディスク３０に物理的に書き込まれていることが保証される。ディスク３０にページを書き込むことは、フリーバッファスペースを生む(そのスペースはライトページで占有される)。それ故、一旦、ページがディスク３０に書き込まれると、ディスクキュー３２内のライ

トページのデッドラインだけでなく、ライトバッファプール２８内のすべてのページのデッドラインを緩和することができる。上述したように、可変値d(t)を全体的に修正することにより、ライトバッファプール２８内のページのデッドラインが緩和される。

【００４０】同様に、ディスクキュー３２に対して、ディスク３０へライトページを書き込むとき、次の手順を適用する。現在の時刻をｔとする。

１．ディスクキュー３２をスキャンする。

２．すべてのライト要求をキュー３２に位置させる。

３．ディスクキュー３２内の各ライト要求p_wに対して優先順位$d_w(t_o)$を付与する。(t_oは、$d_w(t_o)$が修正された最後の時刻)

$d_w(t_o) \leftarrow t - t_o + d_w(t_o) + 1/\lambda_w$

このデッドラインの緩和に対する理由は、システム全体の要求を減じるためであり、それ故、リードページの喪失を減じるためである。

【００４１】リード要求R_{dl}の挿入および処理

リード要求R_{dl}の扱いは、本来のＳＣＡＮＲＴプロセスにて何が実行されるのかに似る。新しいリード要求R_{dl}がディスクキュー３２に到着したとき、図４のフローチャートに示されるように、そのプロセスは、その要求を正確なＳＣＡＮの順位に挿入する。最初、そのプロセスは、その位置を、そのページのスキャン順に対応してキュー３２内に位置させる(Ｓ７０)。その要求をその位置に挿入する前に、プロセスは、キュー３２内のすべての要求のデッドラインが違反していないかをチェックする(Ｓ７２)。これは、ディスクキュー３２内に既に位置するリードおよびライト要求の双方に適用する。デッドラインが違反していなければ、新しいリード要求R_{dl}は、そのスキャン順にディスクキュー３２内に挿入される。もし、一つまたはよりおくのデッドラインが違反しているならば、新しいリード要求R_{dl}は、キュー３２内のそのスキャン位置に挿入されない。その代わりに、キュー３２の終端に挿入するよう試みられる。この場合、もし、新しいリード要求R_{dl}が市販していないと決定されれば(Ｓ７６)、キュー３２の終端に挿入される(Ｓ７８)。そうでなければ、リード要求は、喪失したとみなされ、そして、システムによって処理されない(Ｓ８０)。

【００４２】ここでの可能性のある１つの最適活用は、デッドラインを侵害することなしにディスクキュー３２の末尾に位置を替えることができるキュー３２のライト要求を検索をすることである。リード要求が、デッドラインに至っているリード要求のスキャンオーダー中の新しいリード要求に適応する場合のみ、リード要求のこういった位置替えが、実行される。現在、ライト要求のデッドラインを概算する第２の方法は、上述された方法の処理を表にするディスクの実行を発達させるものとして発表されている。

【００４３】以下の記載を目的として、ディスクキュー３２はＤで示され、キュー３２の位置ｉでの要求（リード又はライト）はＤ［i］で示され、さらにキュー３２の先頭の要求はＤ［0］で示される。ライト要求W_nが位置ｎでのディスクキュー３２に挿入されている、即ちＤ［ｎ］＝W_nを仮定する。ディスクキュー３２の位置０～ｎ－１の数量は、ＮｕｍＷｒｉｔｅｓ（ｎ－１）で示される。ＮｕｍＷｒｉｔｅｓ（ｎ－１）は、ディスクキュー３２のW_nの前のライト要求の数量を示していて、W_nが表に書き込まれるのに先行する（システムによって命令された際のデッドラインの衝突が原因で、何回かのライト要求の再編成が発生するまで）。

【００４４】１のライト要求の処理は、ページがディスク３０に書き込まれること、及びライトバッファープール２８でいくらかのバッファー空間を開放することを含んでいる。結果、例えばW_oであるライト要求がディスクスケジュール処理で処理された場合、このことで$1/\lambda_w$までの全ての未決定のライト要求のデッドラインを緩和する。我々はこの事実を利用して、ディスクキュー３２にライト要求を挿入する時にライト要求のデッドラインをより優れて（より効率的に）予測する。ライト要求Ｗｎが、ディスクキュー３２の位置ｎ、時間ｔで、ディスクキュー３２に挿入されることを仮定する。W_nの挿入に関して、我々は以下の方法でW_nのデッドラインを計算する。

$d(t) = t + (n_r(t) + \mathrm{NumWriters}(n-1))/\lambda_w$

【００４５】これを本発明の第１の実施例で与えられたW_nのデッドラインを計算する公式と比較する。新しい公式はディスクキュー３２のW_nに先行するライト要求の数量を考慮するものであり、ライト要求に関してより現実的で、緩和されたデッドラインを提供する。このことによって、ロストリード要求の数量の減少に関してシステムの混雑が減少されることが望まれる。少ない要求が原因で、このことが緩和されたデッドラインに関するライト要求に課せられる。ライト要求のデッドラインの計算を変形したので、ディスクスケジュール処理の幾つかの密接な関係と変形が発生する。一度システムがライト要求を処理すると（ディスク３０に物理的にページを書き込むまで）、ライト要求のデッドラインをもはや更新する必要はない。その結果、ライト要求を処理するまで、さらなる更新処理は必要ない。他方、ライト要求が位置づけられるごと又はシステムによって新しい位置に再配置されるごと、例えばライト要求をキュー３２の先頭に移動させた場合、ディスクキュー３２中のライト要求のデッドラインは再調整を必要とする。

【００４６】R_{nn}リード要求の挿入と処理

リード要求のR_{nn}カテゴリーは、ビデオを編集する応用例に役立つ。R_{nn}カテゴリーは以下の特性を有する。

１．　R_{nn}にはリアルタイムデッドラインがない。この

ため、別の要求よりも長い遅延を提供することが可能になる。

２．　デッドラインが無いので、R_{nn}リード要求は無制限に待機できない。将来いつか、R_{nn}リード要求は完了されなくてはならない。

３．　システムロード等のおかげでR_{nn}はロストできない。システムロードにも関わらず、R_{nn}リード要求は、時間的なある地点においてシステムによって完了させられなくてはならない。リードページは編集される候補であるから、ユーザーはそれらの全てを得るべきである、即ち処理は、R_{nn}がリード時間中にロストしないことをユーザーに保証する。

【００４７】W_{nn}ライト要求に関して、デッドラインの不足のおかげでW_{nn}ライト要求は、R_{dl}よりも低い優先順位としてみなされる。同様にデッドラインの不足のおかげで、R_{nn}リード要求は、R_{dl}リード要求よりも低い優先順位を有するものとしてみなされる。しかしながらライト要求と比較して、R_{nn}リード要求はバッファー問題がないので（書き込まれるページの空間がある限り）、我々は、R_{nn}もまたライト要求よりも劣る優先順位としてみなすことができる。従ってシステムは、表への書き込み関してR_{nn}リード要求に最下位の優先順位を与える。結果、システム中に未決定のW_{nn}又はR_{dl}要求が無い場合のみ、R_{nn}リード要求はディスクキュー３２に挿入される。

【００４８】処理を実行中、以下の断定が適用できる。

$\forall t \forall pw : dw(f) = d(f)$

この断定は、以下のように言い直すことができる。先のセクションで記載されたようにシステムによって実行された割り当てのデッドライン、及びそれらの維持が与えられているので、処理実行中は常時、各ライトページのデッドラインは、別の全ライトページのデッドラインに等しい。ここで、上記断定が正しいと仮定すると、システムの実行の影響を受けることなしに処理は有意に単純化される。言い換えると、同じ数値を有するマルチプルコピーのデッドラインを更新する代わりに、我々は１つのコピーのデッドラインのみを維持し、かつそのデッドラインを全てのライトページ要求に利用する。実際には我々がディスクキューに存する全ライトページの包括的な１つのデッドライン$d(t)$のみを維持することを必要とすることを上記断定は包含する。結果的にディスクにページを書き込むまで、$d(t)$に限り更新される必要があり、かつライトページのデッドラインを更新する為にディスクキュー中のライトページを走査する必要はない。

【００４９】処理の実行中、以下の断定も適用される。

$\forall t \geqq t_0 : d(t) = t_0 + (N_w + N_{serviced}(t)) / \lambda_w$

t_0は処理の実行を開始した時間であり、N_wはライトバッファープール２８が適合することができるページの総数量であり、さらに$N_{serviced}(t)$は時間t_0から時間ｔまででディスク３０によってサービスされた（書き込まれた）ライトページの総数量である。時間$t_0 = 0$で処理が開始されたと仮定すると、

$\forall t \geqq 0 : d(t) = (N_w + N_{serviced}(t)) / \lambda_w$

【００５０】以上に詳述した新しいリード／ライトスケジュール処理の重要な１つの長所は、デッドライン及びリード要求とライト要求の配置に関してリード／ライトスケジュール処理が、リード及びライト要求の両方を同一の方法で可能な限り多くスキャンオーダーで処理することである。ビデオサーバー環境でディスクリード要求とライト要求とを同時に保持する処理が提供される。ライト要求がビデオクリップ編集又は処理が許可されている映画の結果である一方、リード要求は動画を見ることの結果である。動画を見ることのリアルタイムの要求のおかげで、リード要求がロストしたとみなされない場合、リード要求は一定のデッドライン内で完了されなくてはならない。ディスクに書き込まれるべきデータがメインメモリーバッファーに蓄えられるので、臨界リード要求が処理されるまでライト要求が延期されることができる。しかしながら、妥当な遅延で不確定な延期の可能性なしに、ライト要求は処理されなくてはならない。このことは、メインメモリーライトバッファーの制限された寸法である物理的拘束によるものである。新しい処理はリード要求及びライト要求の両方をスケジュールし、発表されているデッドラインに合致しないディスクリードの数量を最小化し、さらに不確定な延長及びディスクライトケースの大きなバッファー寸法を除外する。

【００５１】

【発明の効果】以上の説明から明らかなように、本発明によれば、ビデオ・オン・デマンド要求により同時にビデオオーサリング及び編集を処理できるように、ビデオサービスの機能拡張を意図したものである。この新たな環境下では、或るユーザは既存のビデオストリーム（例えば、映画）を要求する。ビデオストリームの観点からは、ビデオ編集及びオーサリングがリード／ライトアプリケーションであるのに対して、ビデオ・オン・デマンドはリードオンリーアプリケーションであることが大きな相違点である。この相違は、ビデオサーバに於けるプロセス設計に強い影響を及ぼす。これらの諸目的は、ビデオサーバに於ける同時ディスクリード及びライト要求をサポートすると共に、バッファプールに於ける有用なスペース量に基づいてライト要求に対するデッドラインを割り当てるステップと；リード要求と共通ディスクキューに於ける前記要求を挿入するステップと；そして、キューの先頭に来る前記リード及びライト要求を処理するステップを有する方法を提供することによりを達成出来る。これらの諸目的は、ビデオサーバに於ける同時ディスクリード及びライト要求をサポートすると共に、リ

19

ード及びライト要求を一次的に保存するメモリバッファプールと；メモリバッファプールからリード及びライト要求を受け取る共通ディスクキューと；そして、リード及びライト要求、少なくとも一つのディスクを有するファイルサーバ；ライト要求がデッドラインに割り当てられるディスクキューに於けるリード及びライト要求を挿入する位置を決定する制御手段を有するビデオサーバアーキテクチャを提供することによりを達成出来る。

【００５２】本発明の更なる適用範囲は、以下の記述より明らかになる。しかしながら、詳細な記述及び具体的な事例は発明の好ましい実施例を示す為であって、本技術の熟練したものであれば、そのような記述や事例から本発明の思想及び範囲に属する様々な変形や改良を読みとれることは明白である。このように本発明は記載されていて、同一のものが多くの方法で変形されてもよいことは明白である。こういった変形物は、本発明の分野と

20

意図から離れたものとみなされず、当業者にとって明らかな全ての変形は、以下の請求項の分野に含まれる。

【図面の簡単な説明】

【図１】　本発明に関連して用いられるビデオサーバのアーキテクチャをの模式図である。

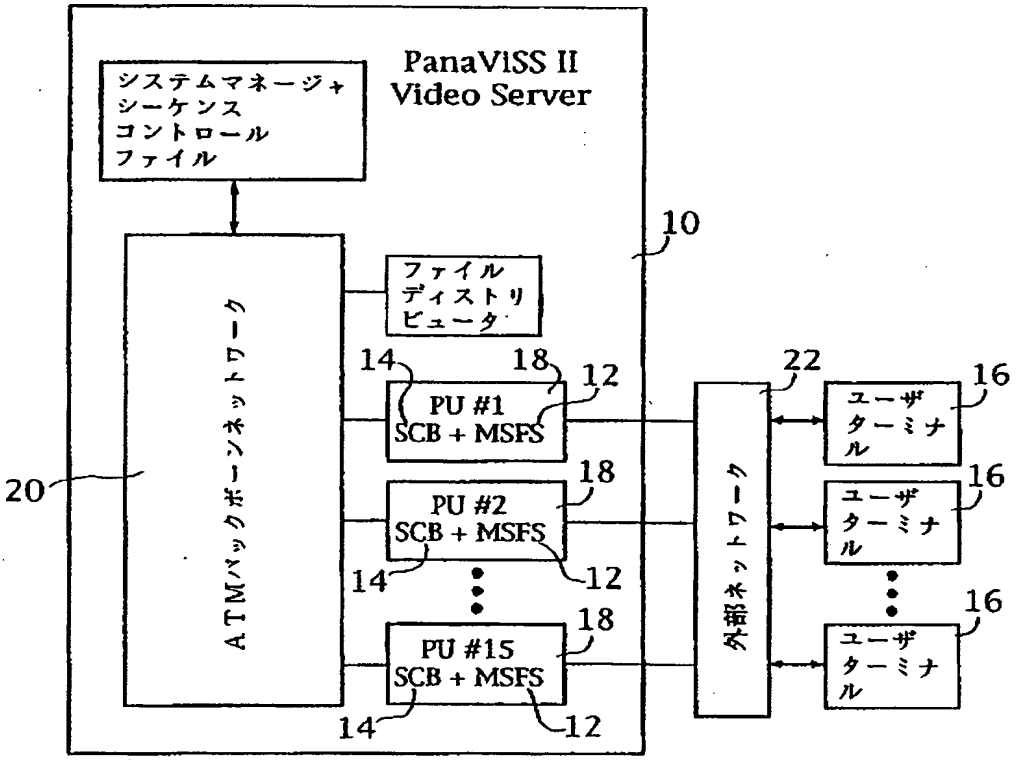
【図２】　本発明に係るリード及びライト要求のデータフロー図である。

【図３】　本発明の原理に基づいてライト要求をディスクキューに挿入する位置の決定に用いられるプロセスを説明するフローチャートである。

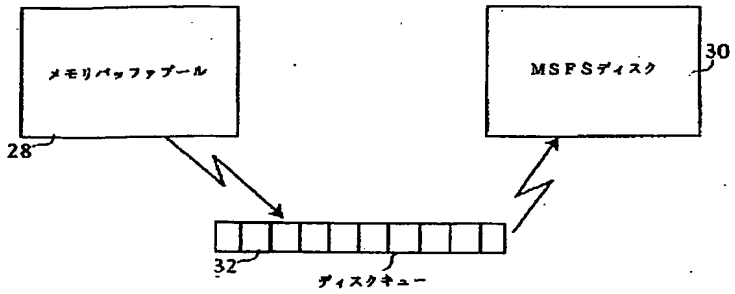
【図４】　本発明の原理に基づいてＲａｉタイプのライト要求をディスクキューに挿入する位置の決定に用いられるプロセスを説明するフローチャートである。

【図５】　（Ａ）及び（Ｂ）で併せて、発明に用いられる技術を俯瞰するブロック図である。

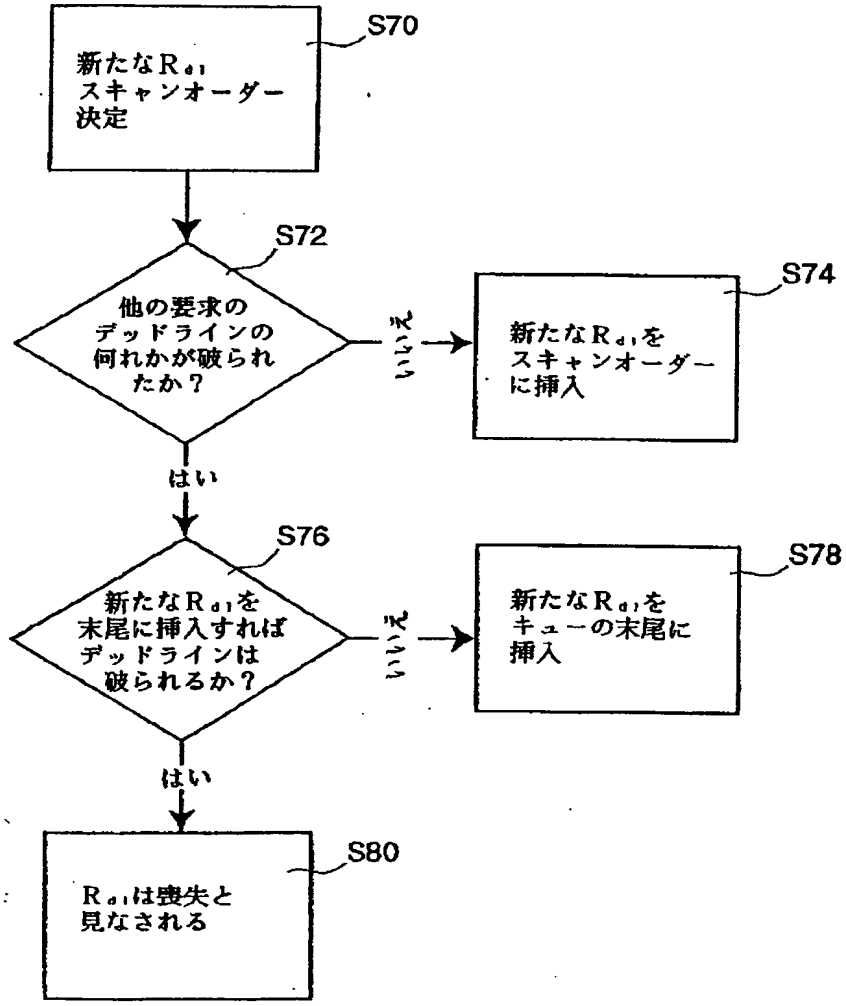
【図１】



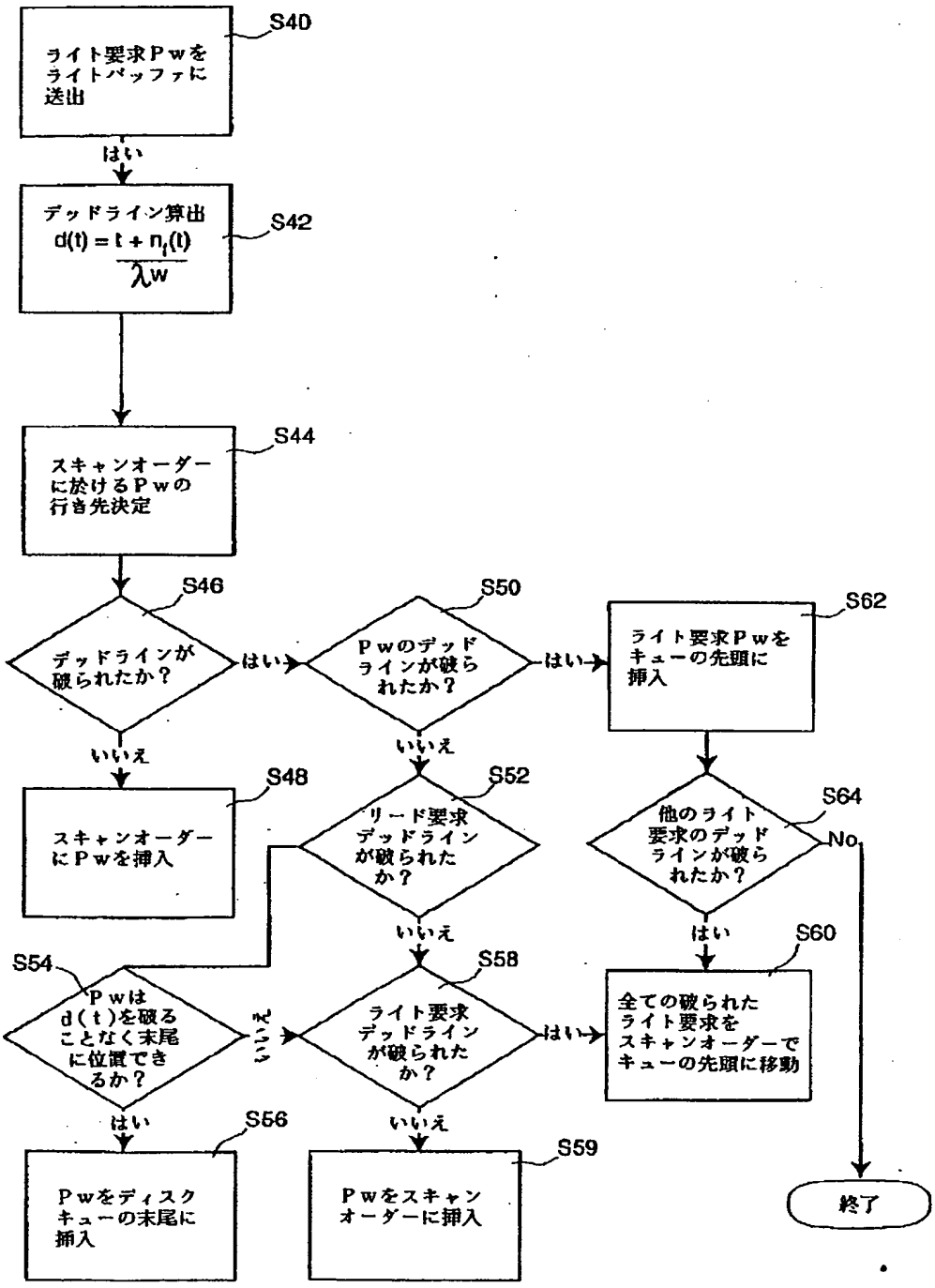
【図２】


メモリバッファプール
28
MSFSディスク
30
32
ディスクキュー

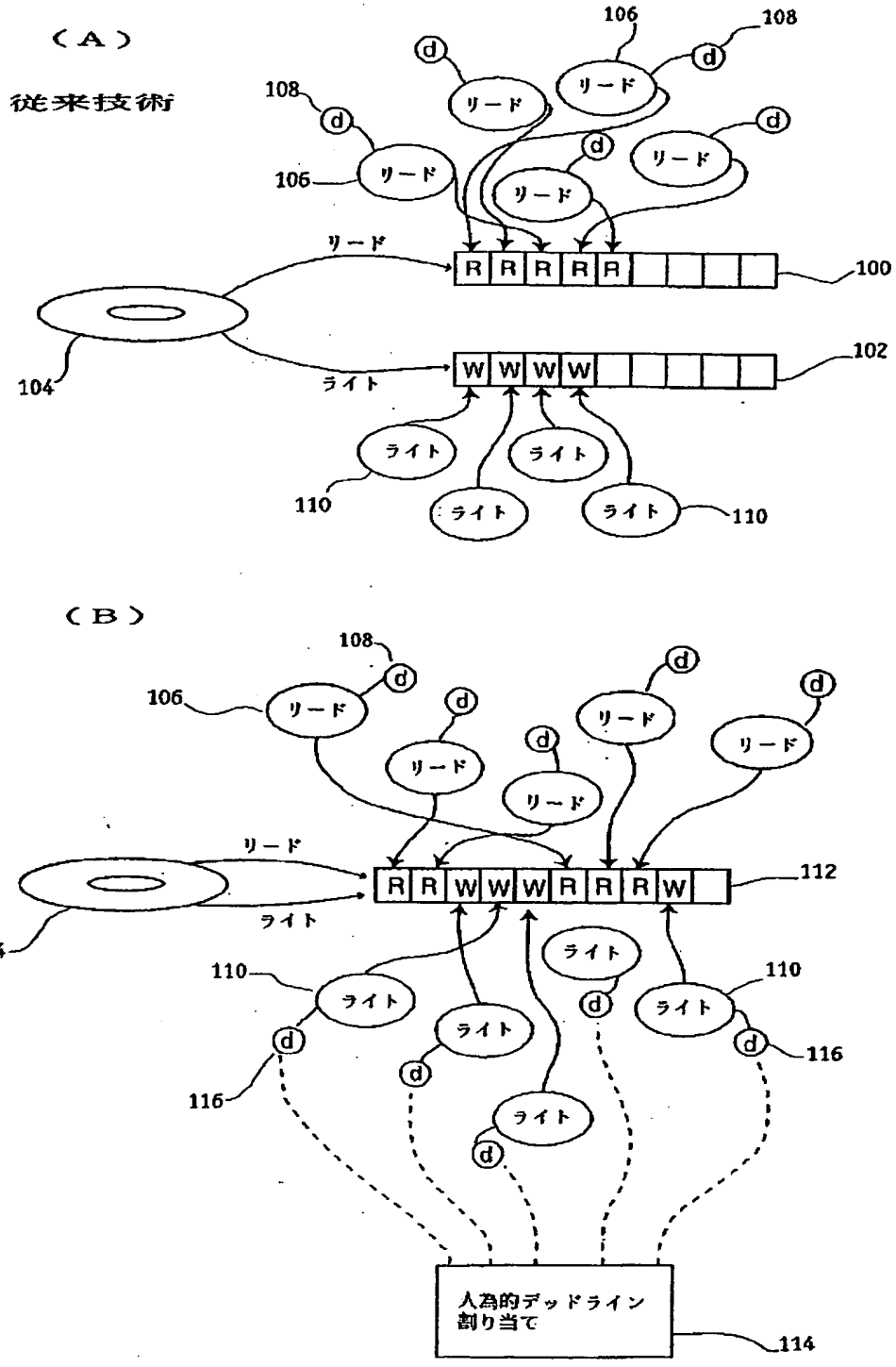
【図４】


S70
新たなRd1
スキャンオーダー
決定
S72
他の要求の
デッドラインの
何れかが破られ
たか？
いいえ
S74
新たなRd1を
スキャンオーダー
に挿入
はい
S76
新たなRd1を
末尾に挿入すれば
デッドラインは
破られるか？
いいえ
S78
新たなRd1を
キューの末尾に
挿入
はい
S80
Rd1は喪失と
見なされる

【図３】


S40
ライト要求Ｐｗをライトバッファに送出
はい
デッドライン算出
$d(t) = t + \frac{n_t(t)}{\lambda w}$
S42
S44
スキャンオーダーに於けるＰｗの行き先決定
S46
デッドラインが破られたか？
はい
いいえ
S48
スキャンオーダーにＰｗを挿入
S50
Ｐｗのデッドラインが破られたか？
はい
いいえ
S52
リード要求デッドラインが破られたか？
いいえ
S54
Ｐｗはｄ(ｔ)を破ることなく末尾に位置できるか？
いいえ
はい
S56
Ｐｗをディスクキューの末尾に挿入
S58
ライト要求デッドラインが破られたか？
はい
いいえ
S59
Ｐｗをスキャンオーダーに挿入
S62
ライト要求Ｐｗをキューの先頭に挿入
S64
他のライト要求のデッドラインが破られたか？
No
はい
S60
全ての破られたライト要求をスキャンオーダーでキューの先頭に移動
終了

【図５】


（Ａ）
従来技術
106
108
d
リード
108
106
リード
リード
R R R R R
100
104
ライト
W W W W
102
ライト
ライト
110
ライト
ライト
110
（Ｂ）
108
106
リード
リード
リード
リード
リード
R R W W W R R R W
112
リード
ライト
104
110
ライト
ライト
ライト
110
116
116
人為的デッドライン
割り当て
114

フロントページの続き

(72)発明者 ラファエル・アロンソ
アメリカ合衆国08512ニュージャージー州
クランベリー、ブレントウッド・レイン９
番